

VCSQuickStart(V1.0, March 19th, 2001)

Notations	Examples	Altering the Template
<pre>representsaslab/transientvariable xisthevcsanobject(i.e.x=vcsvs.init())</pre> <p>Starting up</p> <pre>x=vcsvs.init()#initiatesvcswindow x.plot(s)#Plottheabsan return "canvas" objectthatcanbealtered Tochangeorientation: x.open(#popupsupvcswindow x.page()#togglebetweenportraitandlandscape mode x.portrait()#Setthewindowtoportraitmode x.landscape()#setthewindowlandscape mode x.geometry(width,height,xOff,yOff)#tospecifyyourownwindowsize</pre> <p>Defining VCS objects and using them</p> <pre>Availablegraphicmethodssare: 'contbins':isofill,'isoline','vector','xvsy','outfill','outline','scatter','xysy','yxvsx', 'contlines':isofill'methodisavailable: x.show(isofill)</pre> <p>Toretrievegraphicmethod,e.g.boxfillmethod'quick': box=x.getboxfill('quick')</p> <p>Tocreateanewgraphicmethod,e.g.isolinemethodnamed'new'from'quick': iso=x.createisoline('new','[quick]'):#quickisoptionalfornopassedtenthedefault issued</p> <p>Toplotabusinggraphicmethodobject:e.g."iso"objectandslabrx.plot(iso)</p> <pre>orx.plot(iso,s)orderdoesntmatter AnothertypeofVCSobjectis'templ' at'e.i.,where thi ngsareplottedinthes canvas(seespecialsectionformoreinfo) impl=x.gettemplate('quick')orimpl=x.createtemplate('new','[quick]') Youcanplotaspesifictemplate,gstat's"withgraphicmethod"iso'inthe template'impl'rx.plot(s,iso,tm,ph)inanyorder Otherwiseobjectsexists,theyrepresentsspecificationsusedbygraphic template,theseobjectsare: 'line','marker','fillarea','textorientation'and'textable' Onceagainyoucanreadthemwiththegetcreatenetworones e.g.:fi=x.createfillarea('new') Alternativecolormapcanbeused:x.setcolormap()</pre> <p>Thesecolormapcanbealtered:e.g.tochangethered,green,bluevaluesofcolor20: x.setcolor20(20,red,green,blue)</p> <p>Altering VCS Objects</p> <pre>vcsvs.object.list()list themodifiableattributesof"vcsobject"andtheirvalues seerefencemanualfordetailsontheartributeandtheirfunctions</pre> <p>Outputting</p> <pre>tothescreen:x.plot(s,[method],[template]) totheprinter:x.printer([printname]) toplottedinthebackground nd:atthekeywordbg=1toyourplotcommand tofile: postscript:x.postscript(filename.ps],[orientation]),orientationis!'/p' eps: x.eps('filename.ps',[orientation]),orientationis!'/p' gif: x.gif('filename.gif'),[append],[orientation]),append:a!/r' gml: x.gml('filename.gml'),[append]) raster: x.raster('filename.ras'),[append])</pre> <p>Avcsvobjectcanbesavedaseitherpythonscriptorvcsvscriptbyusingits'script' function:iso.script('filename'),vcsvscript willbesavedif filenameendswith'.scr'</p>	<p>TheVCSSmodulecomeswithsomehandyfunctionalities:</p> <pre>min,max=vcsvs.minmax(slaborlistofslabs)returns theminandmax levs=vcsvs.mkvelevels(n1,n2,[rl]).returnalistoflevelsgoingfromn1ton2and spanningminintervals(i.e.levshash+1values),thedefaultvaluefornis10 intervals</pre> <p>levs=mkscale(l,n1,n2,[lmax],[lzero]).returnalistoflevelscontaining landl2, withamaximumoffmax values.Zerodeterminesifzerocarbeavelevel (zero-l,default).has tobelevel(zero-2)orcannotbelevel(zero- 1)</p> <p>cols=vcsvs.getcolors(levs,[cols=range(16,240)],lwhite-240)[split=1].returnasetof colorspanningthelist'cols'.If"split" is in2halvesonefor>0values,moreoveritgoesfrom <0to>0itwillbeittothecolor"white"(default=241,i.e.White)</p> <p>lbls=vcsvs.mklabelsls(levs,output='dic')returnadictionaryco ntaining "nicely" rewrittenvaluesforthefilexsls(0.00000001 ->1E -8,ifoutputisnotstring withdicthenasimplisistumed</p> <p>Playing with the levels/colors/attributes</p> <pre>Let'sassumewe wanttodefinelithelevelsfordisplaying,sayfrom every5.withthextensionarrows.Herelishowtодоifferentgraphicmethods: Thereafterthelevelswantedaredinthe"levs"listandthelabelsdictionarryin"lbls" andthecolorsarein'cols'</pre> <pre>box=x.createboxfill('new') box.level_1_=levs[0] box.level_2_=levs[1] box.ext_1_=y' box.ext_2_=y' box.legend=lbls x.plot(s,box)</pre> <p>Boxfill</p> <pre>Isosfill</pre> <pre>iso=x.createisosfill('new') levs.append(1,E20) #tohaveanextensionarrowontheleft levs.insert(0,1,E20) #tohaveanextensionarrowontheright iso.level=levs #setsthelevels iso.fillarea.colors=cols x.plot(s,iso)</pre> <p>Isoline</p> <pre>mx,mm=x.mxmn(y) levs.append(1,E20) levs.insert(0,1,E20) iso.level=levs iso.line=dash ,orlistoflinetypes x.plot(s,iso)</pre> <p>yyvs(xyyxxoryyyvy) 2Dplot(y(x))</p>	<p>To create a vcs template object:tmp=x. createtemplate('new')</p> <p>The data area is in: tmp.data. It is represented by the 4 edges: s1->x2andy1->y2, the values are in: tmp.data. For example if you read data for 25x675% of the page horizontally and from 30% to 67% vertically: tmp.data.x1=.25,tmp.data.x2=.75,tmp.data.y1=.3,tmp.data.y2=.7 The box around the data is determined by box1 therefore we should do: tmp.box.x1=.25,tmp.box.x2=.75,tmp.box.y1=.3,tmp.box.y2=.7 the Finally the horizontal tick marks are set by xtic1, mark and xtic1 (xtic12)io seconds, and the labels by xlabel1 (l2).</p> <p>tmp.xtic1.y= .29,tmp.xtic1.y2= .3,tmp.xlabel1.y=.27</p> <p>Setting the vertical tick marks are set by xtic1 (xtic12)io tmp.ytic1.x1=.24,tmp.ytic1.x2=.25,tmp.xlabel1.x=.22</p> <p>Note that actual values of the labels are set in the vcs method, for example on iso.xticlabel1 = dictionary of location/names</p> <p>Finally any attributes of the template can be set on using the priority attribute, for example to turn the gendiff:tmp!.legend.priority=0</p> <p>Remember that if the element of the template plate that you can set is accessible using tmp.list() function. Also if you know which element you wish to set less exhaustive can be obtained by using the list function of this tmp.data.list() will have attribute of data "zone" that you can set.</p>